Formale Sprachen und Komplexität, SS 18,
Prof. Dr. Volker Heun

Übungsblatt 7

Abgabe: bis Mo. 18.06.2018 8 Uhr

Institut für Informatik **Ludwig-Maximilians-Universität München**
Prof. Dr. Volker Heun

Formale Sprachen und Komplexität, SS 18
Übungsblatt 7

Abgabe: bis Mo. 18.06.2018 8 Uhr

Nach Bearbeitung dieses Übungsblattes sollten Sie:

	Check
Informell beschreiben können, welche Funktionen eine Turingmaschine durchführt und mit welchem Ziel.	
Die akzeptierte Sprache einer gegebenen Turingmaschine bestimmen können.	
Eine komplexe Turingmaschine zum Erkennen einer formalen Sprache konstruieren können.	
Eine komplexe Turingmaschine zur Berechnung einer Funktion auf den natürlichen Zahlen konstruieren können.	
Eine Turingmaschine als Komposition von kleineren Turingmaschinen konstruieren können.	

Diese Ziele sind wichtige Hinweise für die Klausur!

Aufgabe 7-1 schriftlich bearbeiten

Turingmaschine zum Berechnen einer Funktion auf Zahlen

Sei *post*2 die Nachfolger-nachfolger funktion auf $\mathbb{N}$, also

$$\begin{aligned} post2 : \mathbb{N} \to &\quad \mathbb{N} \\ n \mapsto &\quad n+2 \end{aligned}$$

Sei $\Sigma = \{0, 1\}$ die Menge der Binärziffern. Die Elemente von Σ^* sollen als natürliche Zahlen in Binärdarstellung interpretiert werden. Um keine Ausnahmen behandeln zu müssen, seien führende Nullen erlaubt und ε eine zulässige Darstellung von $0 \in \mathbb{N}$.

Geben Sie eine deterministische Turingmaschine $inc2 = (Z, \Sigma, \Gamma, \delta, r, \_, \{s\})$ an, die *post2* berechnet. Das Blankzeichen ist _ , der Startzustand ist r, der einzige Zustand, in dem M halten kann (und der einzige Endzustand) ist s.

Lösungsvorschlag:

$Z = \{r, n, e, e1, s\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \_\}$

```
r, 0,   r, 0, R    # nach rechts laufen
r, 1,   r, 1, R    # bis Wortende
r, _,   e1, _, L   # dort letztes zeichen überspringen

e1,0,   e1, 0, L   # letzte 0 überspringen, weiter mit übertrag 1
e1,1,   e, 1, L    # letzte 1 überspringen, weiter mit übertrag 1

n, 0,   n, 0, L    # nach links mit Uebertrag 0
n, 1,   n, 1, L    # laesst Binaerziffern gleich
n, _,   s, _, R    # bei Wortanfang Stop

e, 0,   n, 1, L    # nach links mit Uebertrag 1
e, 1,   e, 0, L    # veraendert Binaerziffern
e, _,   s, 1, N    # Bei Wortanfang Stop
```

Aufgabe 7-2 Turingmaschine zum Erkennen einer formalen Sprache

Betrachten Sie folgende Turingmaschine M:

$M = (Z, \Sigma, \Gamma, \delta, z_0, \_\, E)$ mit
$\Sigma = \{a, b\}, \quad \Gamma = \{a, b, A, B, \_\}, \quad Z = \{z_0, z_a, z_b, \ell, z_{nein}, z_{ja}\}, \quad E = \{z_{ja}\}$

$$
\begin{array}{llcl}
\delta(z_0, a) & = & (z_a, A, R) & \delta(\ell, a) = (\ell, a, L) \\
\delta(z_0, b) & = & (z_b, B, R) & \delta(\ell, b) = (\ell, b, L) \\
\delta(z_0, A) & = & (z_0, A, R) & \delta(\ell, A) = (\ell, A, L) \\
\delta(z_0, B) & = & (z_0, B, R) & \delta(\ell, B) = (\ell, B, L) \\
\delta(z_0, \_) & = & (z_{nein}, \_, N) & \delta(\ell, \_) = (z_0, \_, R) \\
\delta(z_a, a) & = & (z_a, a, R) & \delta(z_b, a) = (\ell, A, L) \\
\delta(z_a, b) & = & (\ell, B, L) & \delta(z_b, b) = (z_b, b, R) \\
\delta(z_a, A) & = & (z_a, A, R) & \delta(z_b, A) = (z_b, A, R) \\
\delta(z_a, B) & = & (z_a, B, R) & \delta(z_b, B) = (z_b, B, R) \\
\delta(z_a, \_) & = & (z_{nein}, \_, N) & \delta(z_b, \_) = (z_{ja}, \_, N)
\end{array}
$$

a) Geben Sie den Konfigurationsablauf auf dem Wort $b\ b\ a\ b\ a$ an.

Lösungsvorschlag:

Zustand	Konfiguration
z_0	_ _ _<b>b a b a _ _ _
z_b	_ _ _ B<b>a b a _ _ _
z_b	_ _ _ B b<a>b a _ _ _
ℓ	_ _ _ B<b>A b a _ _ _
ℓ	_ _ _<B>b A b a _ _ _
ℓ	_ _<_>B b A b a _ _ _
z_0	_ _ _<B>b A b a _ _ _
z_0	_ _ _ B<b>A b a _ _ _
z_b	_ _ _ B B<A>b a _ _ _
z_b	_ _ _ B B A<b>a _ _ _
z_b	_ _ _ B B A b<a>_ _ _
ℓ	_ _ _ B B A<b>A _ _ _
ℓ	_ _ _ B B<A>b A _ _ _
ℓ	_ _ _ B<B>A b A _ _ _
ℓ	_ _ _<B>B A b A _ _ _
ℓ	_ _<_>B B A b A _ _ _
z_0	_ _ _<B>B A b A _ _ _
z_0	_ _ _ B<B>A b A _ _ _
z_0	_ _ _ B B<A>b A _ _ _
z_0	_ _ _ B B A<b>A _ _ _
z_b	_ _ _ B B A B<A>_ _ _
z_b	_ _ _ B B A B A<_>_ _
z_{ja}	_ _ _ B B A B A<_>_ _

b) Geben Sie die von M erkannte Sprache als Teilmenge von Σ^* an.

Lösungsvorschlag:

$\{\omega \in \Sigma^* | |\omega|_a < |\omega|_b\}$ über dem Alphabet $\Sigma = \{a, b\}$

Aufgabe 7-3 schriftlich bearbeiten

Turingmaschine als Komposition von Turingmaschinen

Für diese Aufgabe gelte die Konvention aus der Definition der Turing-Berechenbarkeit: ein k-Tupel von natürlichen Zahlen wird dargestellt als ein Wort aus $\{0, 1, \#\}^*$ bestehend aus den Binärdarstellungen der k Zahlen, die durch je ein # voneinander getrennt sind.

Gegeben seien die Turingmaschinen:

1) *inc* Inkrementiert die erste Binärzahl, die Sie links vom Lesekopf findet, um 1.
   Startzustand: inc_{start}, Endzustand: inc_{stop}

2) *dec* Dekrementiert die erste Binärzahl, die Sie links vom Lesekopf findet, um 1.
   Startzustand: dec_{start}, Endzustand: dec_{stop}

Die beiden Turingmaschinen arbeiten das Eingabewort von rechts nach links ab, und befinden sich danach auf der ersten leeren Stelle vor dem Eingabewort oder auf dem Trennzeichen.
Beispiel: `_ 0 0 1 # 0 0 1<_>` $\overset{dec}{\rightarrow}$ `_ 0 0 1<#>0 0 0 _`

a) Beschreiben Sie informell eine Turingmaschine *add*, die die Summe zweier Binärzahlen berechnet. Nehmen Sie hierzu die Turingmaschinen *inc* und *dec* zu Hilfe.

Lösungsvorschlag:

Die Maschine geht vom Startzustand nach rechts bis zum Trennzeichen.
Dann wird geprüft, ob die zweite Zahl 0 ist.
Wenn ja, gehe in den Endzustand. Wenn nein, gehe bis zum Wortende und starte *dec*.
Nach *dec* wird *inc* gestartet. Nach *inc* wieder in den Startzustand.

b) Konstruieren Sie Ihre Turingmaschine als Kombination der Turingmaschinen *inc* und *dec*.

Lösungsvorschlag:

$Z = \{s, inc_{start}, dec_{start}, z, zf, inc_{stop}, dec_{stop}, ende\}$,
$\Sigma = \{0, 1, \#\}$, $\Gamma = \{0, 1, \#, \_\}$

$\delta(s, \_) = (s, \_, R)$	Vom Start nach rechts bis zum Trennzeichen
$\delta(s, 0) = (s, 0, R)$	nach rechts laufen
$\delta(s, 1) = (s, 1, R)$	bis Trennzeichen
$\delta(s, \#) = (z, \#, R)$	Prüfe, ob die rechte Zahl 0 ist
$\delta(z, 0) = (z, 0, R)$	keine 0, weiter
$\delta(z, \_) = (ende, \_, N)$	zweite Zahl ist 0, wir sind fertig!
$\delta(z, 1) = (zf, 1, R)$	zweite Zahl ist nicht null, weiter bis Wortende
$\delta(zf, 0) = (zf, 0, R)$	weiter nach rechts
$\delta(zf, 1) = (zf, 1, R)$	bis Wortende
$\delta(zf, \_) = (dec_{start}, \_, N)$	hier weiter mit *dec*
$\delta(dec_{stop}, \#) = (inc_{start}, \#, N)$	weiter mit *inc*
$\delta(inc_{stop}, \_) = (s, \_, N)$	Wieder in den Startzustand